Data Documentation

Folders

- raw: z-Tree output, sorted by experiment date. Each file is a different treatment in that experiment, and includes all subject-round decisions and outcomes.
- created: cleaned data, used for analysis

Created data files for analysis

- Final files:
  - alltreatments_subjectlevel.dta
    - Intermediate files include: CVK2m2_subjectlevel.dta, CVK2m3_subjectlevel.dta, CVK4m2_subjectlevel.dta, CVK4m3_subjectlevel.dta, MPK2m2_subjectlevel.dta, MPK2m3_subjectlevel.dta, MPK4m2_subjectlevel.dta, MPK4m3_subjectlevel.dta, alltreatments_m2_subjectlevel.dta, and alltreatments_m3_subjectlevel.dta.
  - monotonicity_behavior_all.dta
    - Intermediate files include: monotonicity_behavior_m2.dta and monotonicity_behavior_m3.dta
- Since all final files include the same variables as intermediate files, we only describe the final file variables.

alltreatments_subjectlevel.dta

Observations are at the m-experiment-treatment-Subject-Round level.

- m: Number of minority voters in each election (treatment parameter), m = {2, 3}. Majority voters are always 4.
- treatment: {CVK2, MPK2, CVK4, MPK4}
- V: Payout if subject's party wins all available seats. V = 400.
- K: Number of open seats. K = {2, 4}.
- experiment: 1-8. We run 8 experiments for each of m = 2 and m = 3.
- Round: Round number (3-17 if first two rounds were practice rounds, 1-15 if not)
- GroupId: Subjects are randomly sorted into one of two groups, changing every round. This prevents learning.
- Team: Party. 1 = minority, 2 = majority
- Subject: Subject ID, unique within (m, experiment).
- VotingCost: randomly drawn voting cost from U[0,100], drawn separately for each subject in each round
- isVote: whether subject voted or not in that round, isVote = {0,1}
- SubjectRoundPayoff: subject's round payoff (earnings from party positions + 100 endowment – voting cost)
- countBigVoters: percent (0-100) of majority voters who voted
- countSmallVoters: percent (0-100) of minority voters who voted
- countBigWinners: number of positions won by majority party
- countSmallWinners: percent (0-100) of positions won by minority party
- countBigPayoff: subject's round payoff, majority party only

- countSmallPayoff: subject's round payoff, minority party only
- ratioTurnout: minority turnout / majority turnout
- isVote_theory: given *VotingCost* and using equilibrium cost cutoffs, should Subjects have voted? (1 if yes, 0 if no)
- countBigVoters_ theory: percent (0-100) of majority voters who voted, if votes were calculated with isVote_theory
- countSmallVoters_ theory: percent (0-100) of minority voters who voted, if votes were calculated with isVote_theory
- countSmallWinners_theory: percent (0-100) of positions won by minority party, if votes were calculated with isVote_theory
- countBigPayoff_theory: subject's round payoff, majority party only, if votes were calculated with isVote_theory
- countSmallPayoff_theory: subject's round payoff, minority party only, if votes were calculated with isVote_theory
- ratioTurnout_theory: minority turnout / majority turnout, if votes were calculated with isVote_theory

monotonicity_behavior_all.dta
Observations are at the m-experiment-treatment-Subject-Vote level. Within a treatment, this file orders each subject's voting decisions in monotonic order of voting cost.

- m: Number of minority voters in each election (treatment parameter), m = {2, 3}. Majority voters are always 4.
- experiment: 1-8. We run 8 experiments for each of m = 2 and m = 3.
- treatment: {CVK2, MPK2, CVK4, MPK4}
- Subject: Subject ID, unique within (m, experiment).
- Team: Party. 1 = minority, 2 = majority
- VotingCost: randomly drawn voting cost from U[0,100], drawn separately for each subject in each round
- isVote: whether subject voted or not in that round, isVote = {0,1}
- mon_order: isVote if subject had voted in a monotonic way. The algorithm to decide this minimizes the number of possible monotonic violations, i.e. the number of changes.
- smallest_mon_violation: The minimum number of monotonic violations for each subject within a treatment.
- min_costcutoff: Since we only have 20 draws per treatment-Subject, we can interpret the cost cutoff as the last cost at which someone voted, or the first cost at which they did not vote. min_costcutoff interprets it as the former.
- max_costcutoff: Since we only have 20 draws per treatment-Subject, we can interpret the cost cutoff as the last cost at which someone voted, or the first cost at which they did not vote. min_costcutoff interprets it as the latter.
- cost_cutoff: Given the constraint that cost_cutoff must be in the interval [min_costcutoff, max_costcutoff], the closest cost_cutoff possible to the theoretical equilibrium (eqm_cutoff).
- mean_cutoff: (min_costcutoff + max_costcutoff) / 2. What we use for our current analysis.
- eqm_cutoff: Theoretical cutoff from calculating equilibrium (external to experiment).